Analysis of Fault Tolerance of a Combining Classifier

Hai Zhao and Bao-Liang Lu

Department of Computer Science and Engineering, Shanghai Jiao Tong University,
1954 Huashan Rd., Shanghai 200030, China
{zhaohai,blu}@cs.sjtu.edu.cn

Abstract. This paper mainly analyses the fault tolerant capability of a combining classifier that uses a K-voting strategy for integrating binary classifiers. From the point view of fault tolerance, we discuss the influence of the failure of binary classifiers on the final output of the combining classifier, and present a theoretical analysis of combination performance under three fault models. The results provide a theoretical base for fault detection of the combining classifier.

1 Introduction

The original purpose of this research arises from two research directions which are different but have some relations.

The first direction is neural network research with fault tolerance. In engineering, the needs of reliability has converted pure hardware concern to hardware-software concern. As a classification model with fault tolerance in essential, the fault tolerance of neural network models has been extensively studied during the past several years [1][2]. The main research approach of fault tolerance is based on the practical fault model or theoretical fault model, e.g., researchers study the shortcut failure of single point with fault tolerance analysis, and then improve the performance or effect through the revised algorithm.

The second direction is concerned with the binarization of multi-class problems which is to be more and more important in the last decade [3][4]. The common ground of those researches is that they studied the integrated output ability of a certain base classifier model, or discussed the total classification error caused by the output combination of some special base classifiers.

Our study focuses on a combining classifier in which component classifier are integrated into a modular classifier with a voting strategy called K-voting in this paper. This combining classifier can also be regarded as a min-max modular neural network[5], which only decompose a K-class classification problem into $\binom{K}{2}$ two-class subproblems and no any further decomposition is performed on two-class subproblems. Throughout this paper, this combining classifier is called K-voting classifier for short.

The basic fault unit for analysis is the binary classification module. Under three fault models, we give the quantitative expression of the ultimate classification effect based on the failure probabilities of binary classifiers.

F. Yin, J. Wang, and C. Guo (Eds.): ISNN 2004, LNCS 3173, pp. 888–893, 2004.

The rest of the paper is organized as follows: In Sections 2 and 3 we briefly introduce K-voting classifier and mathematical notation. Three different fault models will be presented in Section 4. The experimental results and comment on theoretical and experimental results are presented in Section 5. Conclusions of our work and the current line of research are outlined in Section 6.

2 The Combining Procedure of K-Voting Classifiers

Now, we give a simple introduction to combination procedure for K-voting classifier.

we use one-against-one method for task decomposition. Suppose a K-class classification problem is considered, by using one-against-one decomposition method, we get $\binom{K}{2}$ independent two-class problems. The training samples of every two-class problem come from two different classes. K-voting combination rule is defined as: if the outputs of *K-1* binary classifiers in all support the same class ID, then the final classification result is just this class, otherwise, the result is unknown.

3 Mathematical Notation

After binarization, a binary classifier is noted as X_{ij}, which means it has learned from examples in class i and class j. We also take the notation X_{ij} as a stochastic variable with the two-point probability distribution.

For a given testing example S, we denote its class ID as *V(S)* The non-fault output probability p_{ij} of a binary classifier X_{ij} is constrained by

$$p_{ij} = p_{ji} \text{ for } 0 < i, j \leq K, \text{and } i \neq j \tag{1}$$

and X_{ij}'s random guess probabilities for a sample from one non-ij class:

$$P(X_{ij} = i|V(S) = i') = q_{ij} \text{ and } P(X_{ij} = j|V(S) = i') = 1-q_{ij}, \forall i' \neq i \text{ and } i' \neq j \tag{2}$$

The distribution of classes in the number of testing data:

$$\alpha_i \text{ for } 0 < i \leq K, \text{and } \sum_{i=1}^{K} \alpha_i = 1 \tag{3}$$

The correct rate, incorrect rate, and unknown rate obtained by the original K-voting classifier are denoted by

$$P_{OA}, P_{OE}, \text{ and } P_{OF}, \tag{4}$$

respectively. Note that analysis in this paper is based on a simplified assumption: If all occurrence of fault states are regarded as stochastic variants, then these variants are independent with each other.

4 Fault Models

In this section, we will discuss three different fault models and analyse the performance of K-voting classifier under those assumptions.

4.1 Complete Failure Model

We consider the fault model as the output of each binary classifier is complete failure, i.e., fault state is an undefined output state of binary classifier.

Firstly, suppose that the original classification procedure is a faultless procedure, i.e., all binary classifiers can always output the correct results.

In K-voting combination, as to ultimate output class ID i, only those binary classifiers will have the effect on the final output: $X_{mn}, m = i$ or $n = i$. That is, only under such condition $X_{ij} = i, \forall j, 1 \leq j \leq K, i \neq j$, the output can be i.

All the other binary classifiers will have nothing with the final output and fault occurrence is independent, so to the ith class testing sample, the probability of effective output of K-voting classifier will be $\prod_{j=1,j\neq i}^{K} p_{ij}$. After weighting, we get the available output probability of all classes: $P'_{A1} = \sum_{i=1}^{K}(\alpha_i \prod_{j=1,j\neq i}^{K} p_{ij})$.

Finally, according to (4), we get the correct rate, incorrect rate and unknown rate in practical case as follows:

$$P_{A1} = P_{OA}\sum_{i=1}^{K}(\alpha_i \prod_{j=1,j\neq i}^{K} p_{ij}) \quad P_{E1} = P_{OE}\sum_{i=1}^{K}(\alpha_i \prod_{j=1,j\neq i}^{K} p_{ij}) \tag{5}$$

$$P_{F1} = P_{OF}\sum_{i=1}^{K}(\alpha_i \prod_{j=1,j\neq i}^{K} p_{ij}) + (1 - \sum_{i=1}^{K}(\alpha_i \prod_{j=1,j\neq i}^{K} p_{ij})) \tag{6}$$

4.2 Complete Inverse Fault Model

Suppose that the outputs of all binary classifiers can be inversed to irrational output under a certain probability, i.e., to output the rational result under a certain probability, and in other cases, the output is the reverse of the rational judgment according to classification algorithm.

As the same in Section 4.1, we also consider the output performance of K-voting combination while all original outputs of binary classifiers are always correct. because of the independent assumption, at this time, the correct rate of K-voting classifier is given by

$$P'_{A2} = P_{OA}\sum_{i=1}^{K}(\alpha_i \prod_{j=1,j\neq i}^{K} p_{ij}) \tag{7}$$

Because of the fault of inverse, the probability that those binary classifiers can not handle the classification is changed. At this time, for non-i,j classes, the output probability of class ID i by the binary classifier X_{ij} should be revised as:

$$q'_{ij} = q_{ij}p_{ij} + (1 - q_{ij})(1 - p_{ij}) \tag{8}$$

Consider the probability of misclassification of all the samples from class i,

$$\sum_{j=1,j\neq i}^{K} (1 - p_{ij}) \prod_{m=1,m\neq i,m\neq j}^{K} q'_{jm}$$

After weighting, the probability of misclassification for all samples will be:

$$P'_{E2} = \sum_{i=1}^{K}(\alpha_i \sum_{j=1,j\neq i}^{K} (1 - p_{ij}) \prod_{m=1,m\neq i,m\neq j}^{K} q'_{jm}) \tag{9}$$

According to (7) and (9), we may get the unknown rate of K-voting classifier:

$$P'_{F2} = 1 - \sum_{i=1}^{K}(\alpha_i(\prod_{j=1,j\neq i}^{K} p_{ij} + \sum_{j=1,j\neq i}^{K} (1 - p_{ij}) \prod_{m=1,m\neq i,m\neq j}^{K} q'_{jm})) \tag{10}$$

In practice, the unknown rate is very low, so we can omit the output probability of the case that the unknown output is inverted to the correct output or misclassification. But, we can not exclude the case that the inversion causes the final output from the incorrect to the correct. According to Section 4.1, we can get the probability that the correct is inversed from the incorrect as follows:

$$\sum_{i=1}^{K}(\alpha_i \sum_{j=1,j\neq i}^{K} q'_{ij} \prod_{m=1,m\neq i,m\neq j}^{K} (1 - p_{jm}))$$

Ultimately, we get the actual correct rate, incorrect rate, and unknown rate as follows:

$$P_{A2} = P_{OA} \sum_{i=1}^{K}(\alpha_i \prod_{j=1,j\neq i}^{K} p_{ij}) + P_{OE} \sum_{i=1}^{K}(\alpha_i \sum_{j=1,j\neq i}^{K} q'_{ij} \prod_{m=1,m\neq i,j}^{K}(1 - p_{jm})) \tag{11}$$

$$P_{E2} = P_{OE} \sum_{i=1}^{K}(\alpha_i \prod_{j=1,j\neq i}^{K} p_{ij}) + P_{OA} \sum_{i=1}^{K}(\alpha_i \sum_{j=1,j\neq i}^{K} (1 - p_{ij}) \prod_{m=1,m\neq i,j}^{K} q'_{jm}) \tag{12}$$

$$\begin{aligned} P_{F2} = P_{OF} + P_{OA}(1 - \sum_{i=1}^{K}(\alpha_i(\prod_{j=1,j\neq i}^{K} p_{ij} + \sum_{j=1,j\neq i}^{K} (1 - p_{ij}) \prod_{m=1,m\neq i,m\neq j}^{K} q'_{jm}))) \\ + P_{OE}(1 - \sum_{i=1}^{K}(\alpha_i(\sum_{j=1,j\neq i}^{K} p_{ij} + \sum_{j=1,j\neq i}^{K} q'_{ij} \prod_{m=1,m\neq i,m\neq j}^{K} (1 - p_{jm})))) \end{aligned} \tag{13}$$

Because we have omitted the probability of the unknown output inverse to the correct or the incorrect, (11) and (12) will be underestimated, and (13) will be overestimated.

4.3 Pseudo-Correct Output Model

In this subsection, we assume that the binary classifier may not give out the normal output according to rational judgment, but give a random guess according to a certain probability when the binary classifier is in its fault state. Suppose the probability of the output of class ID i under the fault situation of binary classifier X_{ij} is given by:

$$r_{ij} \quad \text{for } 0 < i, j \leq k \text{ and } i \neq j \tag{14}$$

Consider that the outputs of all the binary classifiers are always correct. Then the actual output caused by fault will be

$$p'_{ij} = p_{ij} + (1 - p_{ij})(\frac{\alpha_i}{\alpha_i + \alpha_j} r_{ij} + \frac{\alpha_j}{\alpha_i + \alpha_j}(1 - r_{ij})), 0 < i, j \leq k, i \neq j \tag{15}$$

Therefore, we may convert the pseudo-correct output model to the complete inverse fault model. As mentioned in Section 4.2, according to (8), we obtain:

$$q'_{ij} = q_{ij} p'_{ij} + (1 - q_{ij})(1 - p'_{ij}) \tag{16}$$

Finally, by substituting the p_{ij} with p'_{ij} in (11), (12), and (13), we will obtain the ultimate classification performance of the classifier in this fault model.

5 Computer Simulations and Discussion

We carry out simulations on a practical ten-class classification task. The output of each binary classifier will be disturbed before K-voting combination as a simulation of fault occurrence. The correct rate, the incorrect rate, and the unknown rate is 65.6198%, 28.5537%, and 5.8264% under non-fault condition, respectively. For page limitation, only comparison between the theoretical and the practical performance under pseudo-correct fault model is presented. The experimental results are shown in Table 1. For convenience, non-fault probability of each binary classifier is set to the same value and each r_{ij} is set to 0.5.

The simulation result basically coincides with our theoretical analysis. Also, as we mentioned before, the correct rate and incorrect rate do be underestimated and unknown rate does be overestimated. Under any fault model, we find that:

a) K-voting combination is highly sensitive to the fault of binary classifier.
b) The fault of a K-voting classifier is largely unknown output, instead of the misclassification.

6 Conclusions and Future Work

We have proposed a mathematical model for analysis of the fault tolerance of binarization classification procedure with a quantitative approach in this paper. We also give a better understand of the binarization procedure of the K-voting

Table 1. Simulation results on pseudo-correct model

Reliability	Correct rate(%)		Incorrect rate (%)		Unknown rate(%)	
	Actual	Theoretical	Actual	Theoretical	Actual	Theoretical
50%	5.5372	3.695282	5.4545	1.607959	89.0083	94.696659
75%	20.4545	17.262967	11.6116	7.511781	67.9339	75.225152
90%	41.0331	39.288998	19.8347	17.096155	39.1322	43.614747
95%	51.8595	50.942594	23.4298	22.167083	24.7107	26.890222
99%	62.6446	62.411657	27.3554	27.157713	10.0000	10.430530

classifier. The analysis results give a theoretical base of the influence caused by the fault of binary classifier on the ultimate combination classification result.

In fact, our analysis procedure has no relation with the features of binary classifiers or even the performance of binary classifiers. Therefore, our method has the common sense to some degree. In addition, the fault model we presented, especially the third one, has some value in practice application.

The further work may focus on the direction of fault detection. Through observation on the classification performance before and after fault occurrence, to locate the binary classifier with fault is possible under some prior fault model.

Acknowledgements. The authors would like to thank Mr Yi-Ming Wen, Mr Bin Huang and Miss Lu Zhou for their helpful advice. This research was partially supported by the National Natural Science Foundation of China via the grant NSFC 60375022.

References

1. Bolt, G., Austin J., Morgan, G.: Fault-tolerant Multilayer Perceptron Networks. Department of Computer Science, York University, Hestington, York, UK, TechRep:YCS180 (1992)
2. Hammadi, N.C., Ito, H.: A Learning Algorithm for Fault Tolerant Feed Forward Neural Networks. IEICE Trans. on Information and Systems, E80-D(1) (1997) 21-27
3. Fürnkranz,J.: Round Robin Classification. The Journal of Machine Learning Research, Vol.2 (2002) 721-747
4. Allwein, E.L., Schapire, R. E., Singer, Y.: Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers. Journal of Machine Learning Research, Vol.1 (2000) 113-141
5. Lu, B.L., Ito, M.: Task Decomposition and Module Combination Based on Class Relations: a Modular Neural Network for Pattern Classification. IEEE Transactions on Neural Networks, Vol.10 (1999) 1244-1256